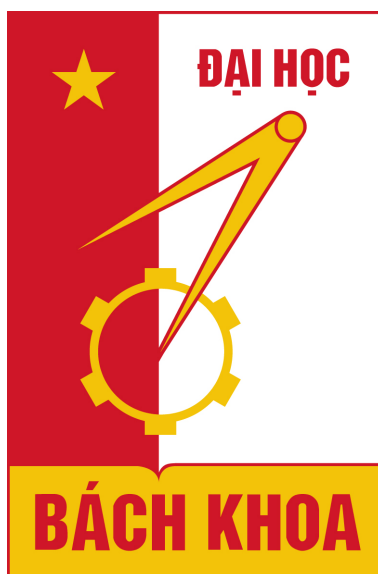


ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
— o0o —



**Thiết kế thuật toán tiến hóa để giải quyết
bài toán TSP**

Môn học: Lập trình hướng đối tượng - IT3103

Giảng viên hướng dẫn: Nguyễn Thị Thu Trang

Mã lớp: 143577

Nhóm 03: Nguyễn Thị Minh Châu - 20214997
Nguyễn Thanh Nhật Bảo - 20210096
Võ Việt Bắc - 20205055
Nguyễn Minh Chiến - 20215000

Mục lục

| | | |
|----------|--|-----------|
| 1 | Khảo sát, đặc tả yêu cầu bài toán | 4 |
| 1.1 | Mô tả yêu cầu bài toán | 4 |
| 1.2 | Biểu đồ usecase | 5 |
| 2 | Thiết kế hệ thống | 6 |
| 2.1 | Biểu đồ gói tổng quan | 6 |
| 2.2 | Biểu đồ lớp chi tiết | 7 |
| 2.3 | Giải thích thiết kế | 7 |
| 2.4 | Package map: | 7 |
| 2.5 | Package genalgorithm.gen: | 7 |
| 2.6 | Package genalgorithm.test: | 8 |
| 2.7 | Package gui: | 8 |
| 3 | Các kỹ thuật lập trình hướng đối tượng đã áp dụng | 9 |
| 4 | Demo chương trình | 10 |
| 5 | Phân công công việc cụ thể | 11 |
| 6 | Kết luận | 12 |
| 7 | Tài liệu tham khảo | 13 |

Lời nói đầu

Lập trình hướng đối tượng (Object-oriented programming, viết tắt: OOP) là một mẫu hình lập trình dựa trên khái niệm "công nghệ đối tượng", mà trong đó, đối tượng chứa đựng các dữ liệu, trên các trường, thường được gọi là các thuộc tính; và mã nguồn, được tổ chức thành các phương thức.

Phương thức giúp cho đối tượng có thể truy xuất và hiệu chỉnh các trường dữ liệu của đối tượng khác, mà đối tượng hiện tại có tương tác (đối tượng được hỗ trợ các phương thức "this" hoặc "self"). Trong lập trình hướng đối tượng, chương trình máy tính được thiết kế bằng cách tách nó ra khỏi phạm vi các đối tượng tương tác với nhau. Ngôn ngữ lập trình hướng đối tượng khá đa dạng, phần lớn là các ngôn ngữ lập trình theo lớp, nghĩa là các đối tượng trong các ngôn ngữ này được xem như thực thể của một lớp, được dùng để định nghĩa một kiểu dữ liệu... OOP được xem là giúp tăng năng suất, đơn giản hóa độ phức tạp khi bảo trì cũng như mở rộng phần mềm bằng cách cho phép lập trình viên tập trung vào các đối tượng phần mềm ở bậc cao hơn.

Ngoài ra, nhiều người còn cho rằng OOP dễ tiếp thu hơn cho những người mới học về lập trình hơn là các phương pháp trước đó. Một cách giản lược, đây là khái niệm và là một nỗ lực nhằm giảm nhẹ các thao tác viết mã cho người lập trình, cho phép họ tạo ra các ứng dụng mà các yếu tố bên ngoài có thể tương tác với các chương trình đó giống như là tương tác với các đối tượng vật lý.

Hiểu được tầm quan trọng và sự hiệu quả của học phần OOP, nhóm được giao nhiệm vụ thực hiện bài tập nhóm nhằm nâng cao, nắm chắc kiến thức cũng như áp dụng vào vấn đề thực tiễn.

Cuối cùng, nhóm 3 xin gửi lời cảm ơn đến cô, anh trợ giảng cùng nhà trường đã tạo điều kiện cho chúng em được học tập và tiếp thu kiến thức bổ ích từ môn học. Xin chân thành cảm ơn !

| Thành viên | Cụ thể |
|-------------------------------|--------|
| Nguyễn Thị Minh Châu (leader) | 41% |
| Nguyễn Thanh Nhật Bảo | 41% |
| Võ Việt Bắc | 15% |
| Nguyễn Minh Chiến | 3% |

Bảng 1: Đóng góp

Chương 1

Khảo sát, đặc tả yêu cầu bài toán

1.1 Mô tả yêu cầu bài toán

Tổng quan: Bài toán du lịch hành khách (TSP) là một bài toán nổi tiếng trong Khoa học Máy tính. Tuyên bố vấn đề khá đơn giản, cho một đồ thị $G = (V, E)$, trong đó V là tập các nút, E là tập các cạnh, có thông tin bổ sung về khoảng cách giữa các nút. Mục tiêu của bài toán là đưa ra tuyến đường ngắn nhất bắt đầu từ V và kết thúc tại V , sao cho mỗi nút được thăm một lần. Dự án này triển khai một Thuật toán Gen (GA) để xác định tuyến đường tối ưu.

Yêu cầu kiến thức cơ bản: Thuật toán, Cấu trúc dữ liệu (không cần kiến thức nâng cao về thuật toán tiến hóa, có thể triển khai dự án này mà không cần kiến thức trước về GA)

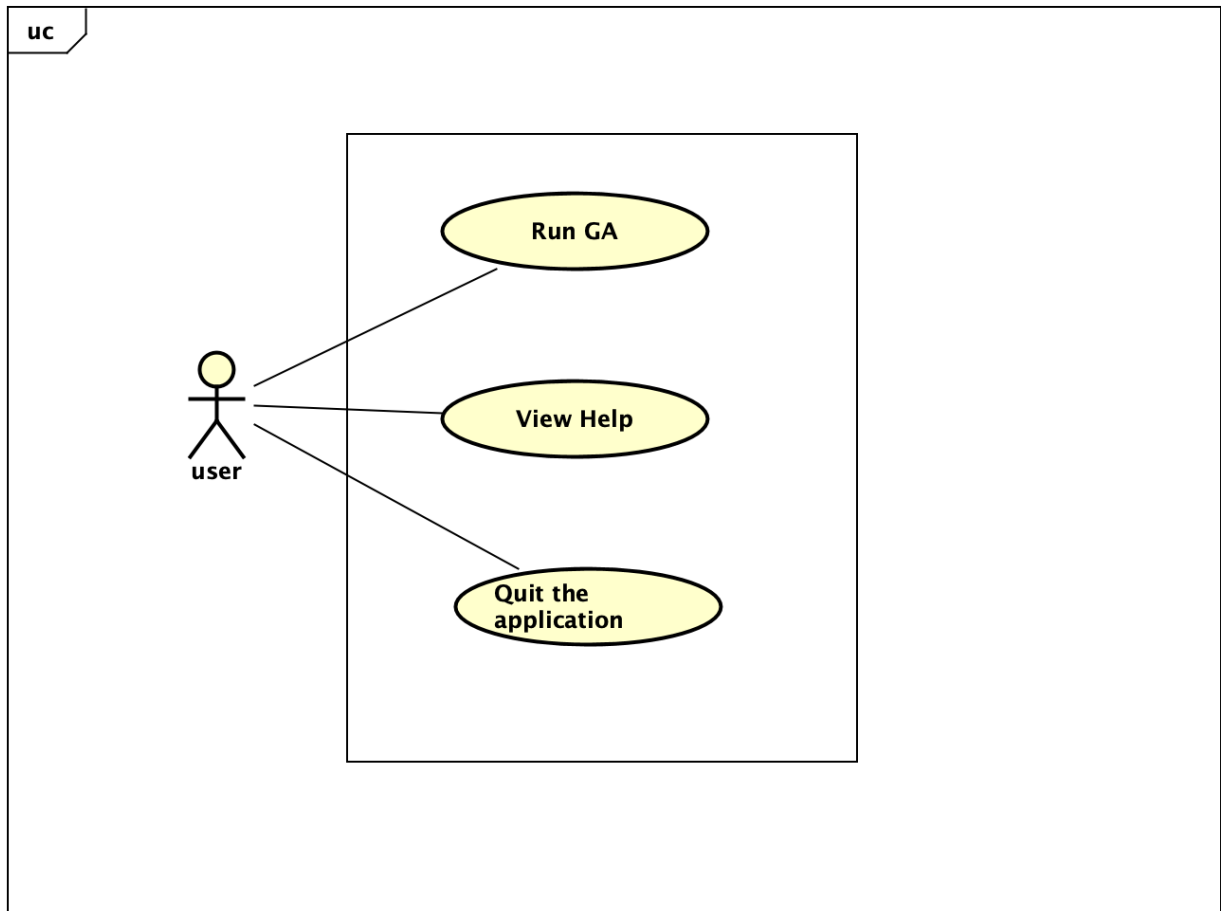
Yêu cầu về giao diện người dùng (GUI):

- Thiết kế:
- Trên menu chính: tiêu đề của ứng dụng, 3 nút cho các thuật toán sắp xếp để người dùng giúp đỡ, giải quyết TSP và thoát
 - Menu Trợ giúp hiển thị cách sử dụng cơ bản và mục tiêu của chương trình
 - Tùy chọn Thoát thoát khỏi chương trình. Nhớ hỏi xác nhận
- Trong bản minh họa
 - Một nút để bắt đầu mô phỏng, cụ thể là "Bắt đầu"
 - Một hình minh họa cho cá thể hiện tại giữ fitness tốt nhất, trong cấu trúc mảng
 - Một hình minh họa cho cá thể hiện tại giữ fitness tốt nhất, trong cấu trúc đồ thị

1.2 Biểu đồ usecase

Giải thích:

- **View Help (Đại diện cho chức năng xem trợ giúp):** Người dùng có thể xem hướng dẫn sử dụng, giải thích các tham số và chức năng của ứng dụng.
- **Run GA (Đại diện cho chức năng chạy giải thuật di truyền):** Người dùng có thể chạy giải thuật để giải quyết vấn đề TSP. Có thể nhập các tham số như số lượng thành phố, kích thước quần thể và số lượng thế hệ.
- **Quit the application (Đại diện cho chức năng thoát ứng dụng):** Người dùng có thể thoát khỏi ứng dụng một cách an toàn sau khi hoàn thành công việc hoặc không muốn tiếp tục.

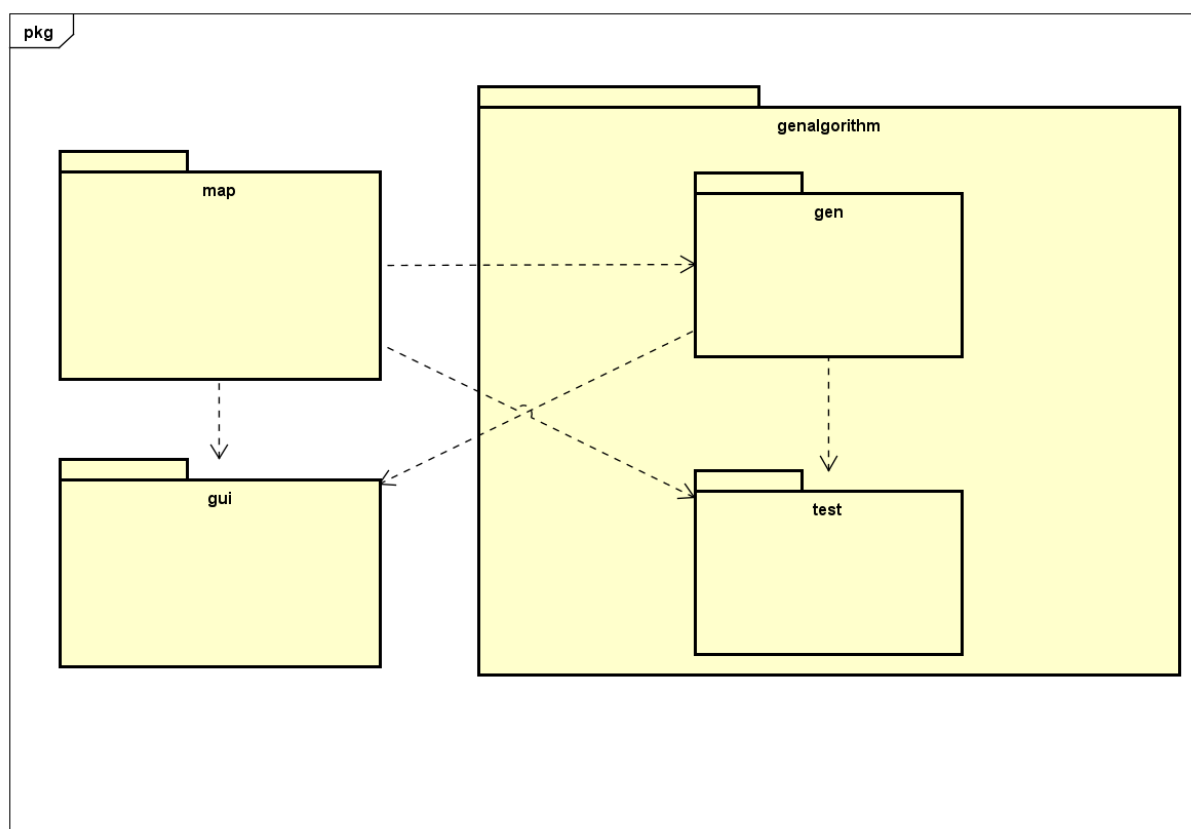


Hình 1.1: Biểu đồ usecase

Chương 2

Thiết kế hệ thống

2.1 Biểu đồ gói tổng quan



Hình 2.1: Biểu đồ gói

- GA: Thực hiện các phương thức liên quan đến quá trình tiến hóa như crossover, mutation và evolution.
- Population: Đại diện cho một quần thể của các tours và cung cấp các phương thức để quản lý và tương tác với quần thể.

2.6 Package `genalgorithm.test`:

Chứa các classes dùng để kiểm thử và chạy thuật toán di truyền.

- TSP_GA: Chứa hàm `runGA` để chạy thuật toán với đầu vào từ người dùng và main để thực hiện việc nhập các thông số từ bàn phím.

2.7 Package `gui`:

Chứa các classes và components liên quan đến giao diện người dùng (GUI).

- TSP_GUI: Class chính quản lý giao diện, sự kiện và tương tác giữa thuật toán và người dùng.
- CityPanel: JPanel tùy chỉnh để vẽ thành phố và các đường nối trên GUI.

Chương 3

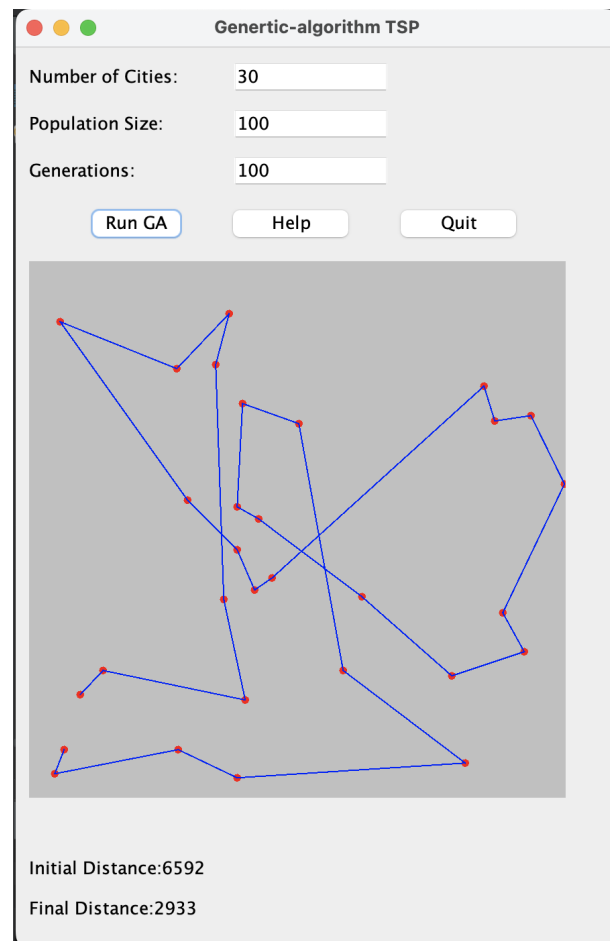
Các kỹ thuật lập trình hướng đối tượng đã áp dụng

1. Tính đóng gói: Các class như City, Tour, và Population có các trường được đặt là private, và các phương thức getter và setter được cung cấp để truy cập và cập nhật dữ liệu. Điều này giúp bảo vệ dữ liệu khỏi sự truy cập trực tiếp từ bên ngoài.
2. Tính kế thừa: Class TSP_GUI được xây dựng trên cơ sở của class JFrame. Điều này thể hiện quan hệ "is-a", trong đó TSP_GUI là một loại JFrame. Class Population là một tập hợp (collection) của các tour, thể hiện mối quan hệ "has-a" với class Tour. Điều này thể hiện tính kế thừa của Population từ Tour và mô phỏng mối quan hệ "is-a".
3. Tính đa hình: Sử dụng Timer trong GUI để tạo đa nhiệm, cập nhật giao diện đồ họa sau mỗi thế hệ của thuật toán mà không làm đóng băng giao diện người dùng.
4. Tính hợp thành: Class TSP_GUI sử dụng sự hợp thành để tích hợp các thành phần như JLabel, JTextField, và JButton. Điều này giúp xây dựng giao diện người dùng bằng cách kết hợp các thành phần nhỏ thành một cấu trúc lớn và dễ quản lý. Class Population chứa một mảng (tours) của các đối tượng Tour, thể hiện mối quan hệ "has-a" thông qua sự hợp thành. Class Population tự nó là một đối tượng độc lập có khả năng duy trì một tập hợp các tours.

Chương 4

Demo chương trình

Video demo



Hình 4.1: GUI

Chương 5

Phân công công việc cụ thể

1. Nguyễn Thị Minh Châu 20214997
 - Tạo lớp City, lớp Tour, lớp TourManagement, lớp Poppulation
 - Làm slide
 - Viết báo cáo
2. Nguyễn Thanh Nhật Bảo 20210096
 - Vẽ general class diagram, detailed class diagram
 - Tạo lớp TSP_GUI, lớp TSP_GA, lớp GA
3. Võ Việt Bắc 20205055
 - Vẽ general class diagram, usecase diagram
4. Nguyễn Minh Chiến 20215000

Chương 6

Kết luận

Dự án bài tập lớn lập trình hướng đối tượng đã cung cấp cho sinh viên cái nhìn tổng quan về cách một dự án hoạt động với phương pháp lập trình hướng đối tượng, cùng với đó là cách quản lý và hoạt động nhóm khi tham gia hoặc tổ chức một dự án công nghệ như thế nào. Sau chương trình học cùng với bài tập lớn, sinh viên chúng em đã có nền tảng kiến thức vững chắc về lập trình hướng đối tượng, một trong những kỹ năng vô cùng quan trọng trong công việc sau này dù có phát triển theo bất kỳ định hướng nào. Ngoài ra những kỹ năng mềm về khả năng làm việc với git, khả năng trao đổi thảo luận nhóm cũng được cải thiện và phát triển rất nhiều

Chương 7

Tài liệu tham khảo

Tham khảo triển khai thuật toán: <https://www.theprojectspot.com>

Tham khảo thiết kế GUI: <https://www.youtube.com/watch?v=94p5NUogClM>

Thư viện swing: <https://zetcode.com/javaswing/>